CLAIMS

WHAT IS CLAIMED IS:

1. A method to manage interactions between applications and a data store, comprising:

(a) concurrently extracting data from a data store into a first queue;

(b) concurrently servicing a plurality of applications with portions of the data from the first queue;

(c) concurrently loading results data into a second queue, wherein the plurality of applications concurrently produce the results data;

(d) concurrently populating the results data into a temporary table; and

(e) merging the temporary table with a data store table of the data store.

2. The method of claim 1 further comprising:

(f) establishing a plurality of first queues, wherein each first queue is associated with a separate processing node that executes a subset of the plurality of applications and each first queue includes the extracted data and concurrently services the subset of the plurality of applications on the separate processing node; and

(g) establishing a plurality of second queues, wherein each second queue is associated with the separate processing node that executes the subset of the plurality of applications and each second queue is currently loaded with results data from the subset of the plurality of applications and concurrently populates a second table, which is merged with the temporary table before performing the merging.

3. The method of claim 1 wherein (b) further includes concurrently servicing a number of the applications from a first processing node and concurrently servicing a remainder of the applications from a second

processing node, wherein the first queue resides on the first processing node.

4. The method of claim 1 wherein (c) further includes currently loading the second queue with portions of the results data acquired from a number of the applications processing on a first processing node and remaining portions of the results data acquired from a remainder of the applications processing on a second processing node, wherein the second queue resides on the first processing node.

5. The method of claim 1 wherein (d) further includes currently populating the temporary table with portions of the results data received from the second queue associated with a number of the applications processing on a first processing node and with remaining portions of the results data received from a different second queue associated with a remainder of the applications processing on a second processing node, wherein the second queue resides on the first processing node and the different second queue resides on the second processing node.

6. The method of claim 1 further comprising (f) concurrently initiating a number of the applications on a first processing node and a remainder of the applications on a second processing node.

7. The method of claim 6 processing further comprising concurrently establishing the processing of (b)-(d) on the first processing node and the second processing node.

8. A method to manage interactions between applications and a data store, comprising:

receiving a query for a data store and an identifier for an application that desires to process the results of the query and update the data store

with application data;

concurrently initiating multiple instances of an application associated with the identifier on multiple processing nodes;

concurrently processing the query and housing the results in one or more application queues residing on one or more of the processing nodes; and

concurrently servicing each of the instances of the application from the one or more application queues.

9. The method of claim 8 further comprising:

concurrently housing the application data in one or more load queues residing on one or more of the processing nodes; and

concurrently populating one or more tables residing on one or more of the processing nodes with the application data from the one or more load queues.

10. The method of claim 9 further comprising merging the one or more tables into the data store.

11. The method of claim 8 wherein the currently initiating further includes determining a total number of the applications to initiate based on configuration data.

12. The method of claim 11 wherein the currently initiating further includes determining which of a number of the applications that are to be initiated on which of a number of the processing nodes based on the configuration data.

13. The method of claim 8 further comprising concurrently synchronizing the application queues and the load queues on the multiple processing nodes when at least some of the processing nodes lack one of the one or

more application queues or one of the one or more load queues.

14. The solution template system of claim 13 wherein the concurrently synchronizing further includes establishing socket based communications between the multiple processing nodes with a Transmission Control Protocol/Internet Protocol (TCP/IP).

15. A data store application management system, comprising:

one or more application queues for servicing one or more applications with results of a query to a data store;

one or more load queues for housing application data produced by the one or more applications; and

a merge utility for merging the application data into a data store table.

16. The system of claim 15 further comprising a configuring utility for determining a total number of the one or more applications.

17. The system of claim 15, wherein the configuring utility initiates a number of the one or more applications, the one or more application queues, and the one or more load queues on separate processing nodes.

18. The system of claim 15, wherein each of the one or more applications concurrently processes the results and produces different portions of the application data.

19. The system of claim 18, wherein each of the one or more application queues and each of the one or more load queues concurrently update while the one or more applications process.

20. A data store residing in a computer-readable medium, comprising:

one or more temporary tables that temporarily house application data produced from concurrently processing applications in response to concurrently provided query results extracted from the data store; and

an application data table that houses application data once the applications have finished producing the application data, and wherein the one or more temporary tables are merged into the application data table.

21. The data store of claim 20 wherein a merge utility merges the one or more temporary tables to produce the application data table once each of the plurality of applications have finished processing the query results.

22. The data store of claim 20 wherein one or more extract utilities perform a query against the data store in order to acquire the query results, which are concurrently consumed by the one or more applications to produce the application data.

23. The data store of claim 22 wherein each of the one or more extract utilities concurrently populate the query results to one or more application queues.

24. The data store of claim 23 wherein each of one or more load utilities concurrently receive portions of the application data from one or more load queues and concurrently populate the portions to the one or more temporary tables.

25. The data store of claim 22 wherein the data store is a least one of one or more databases and a data warehouse.